

SNAP: Symbolic Nuclear Analysis Package A Graphical User Interface for USNRC Analytical Codes



U.S. Nuclear Regulatory Commission

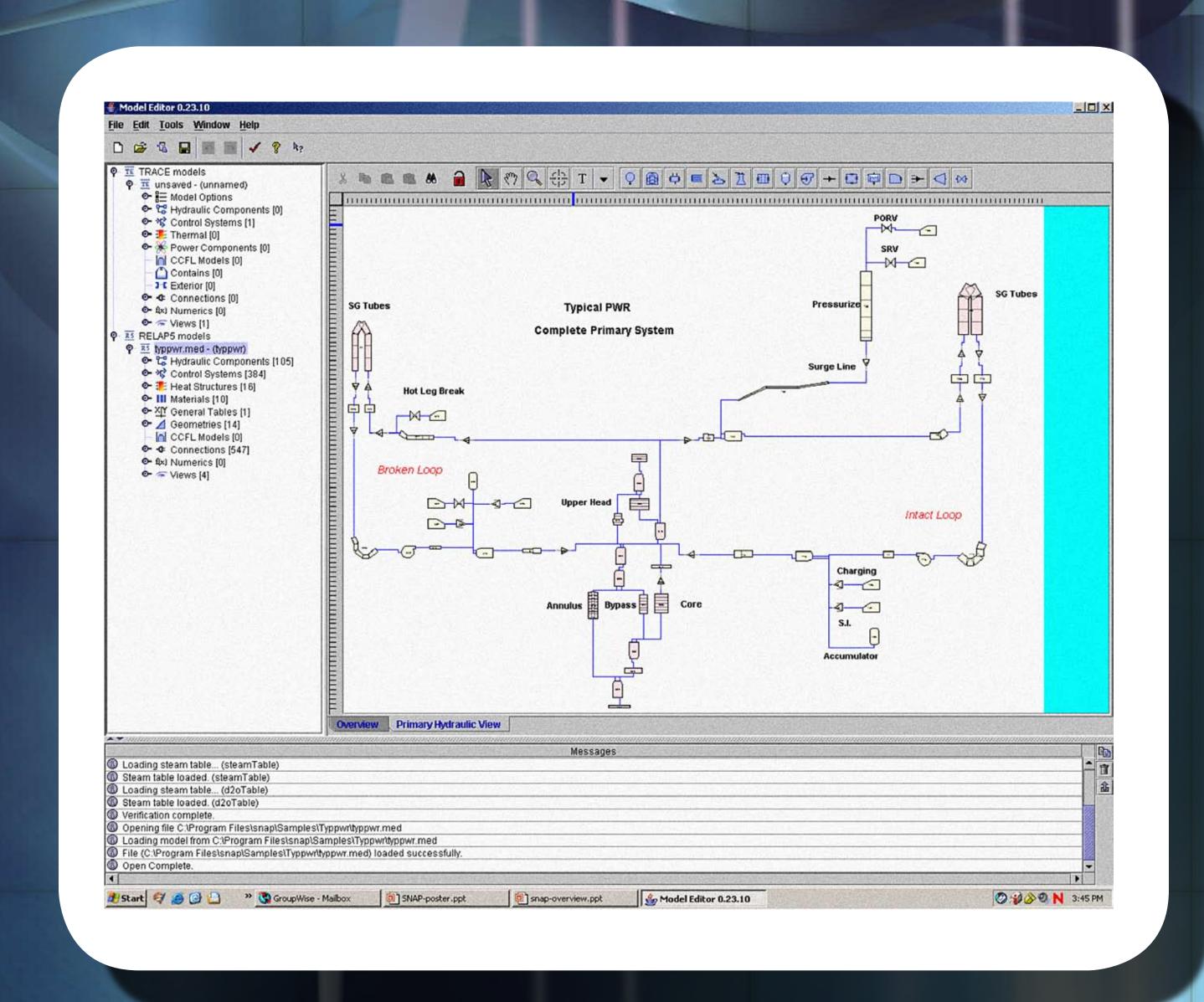
SNAP Symbolic Nuclear Analysis Package

What Is SNAP

- SNAP provides a common graphical user interface for various analytical codes.
- Provides tools to assist analysts in developing input models for the various codes.
- Provides tools to control job execution as well as organize multiple job runs.
- Provides output visualization tools including an animation facility as well as publication quality plotting capabilities.

SNAP Model Editor

- Many analytical tools used by the USNRC use ASCII based input model specifications.
- There are many different input model specification formats.
- The SNAP model editor can read and write the various input files for supported codes.
- Instead of reading through complex ASCII file specifications the analyst can use a simple graphical interface to build an input model.

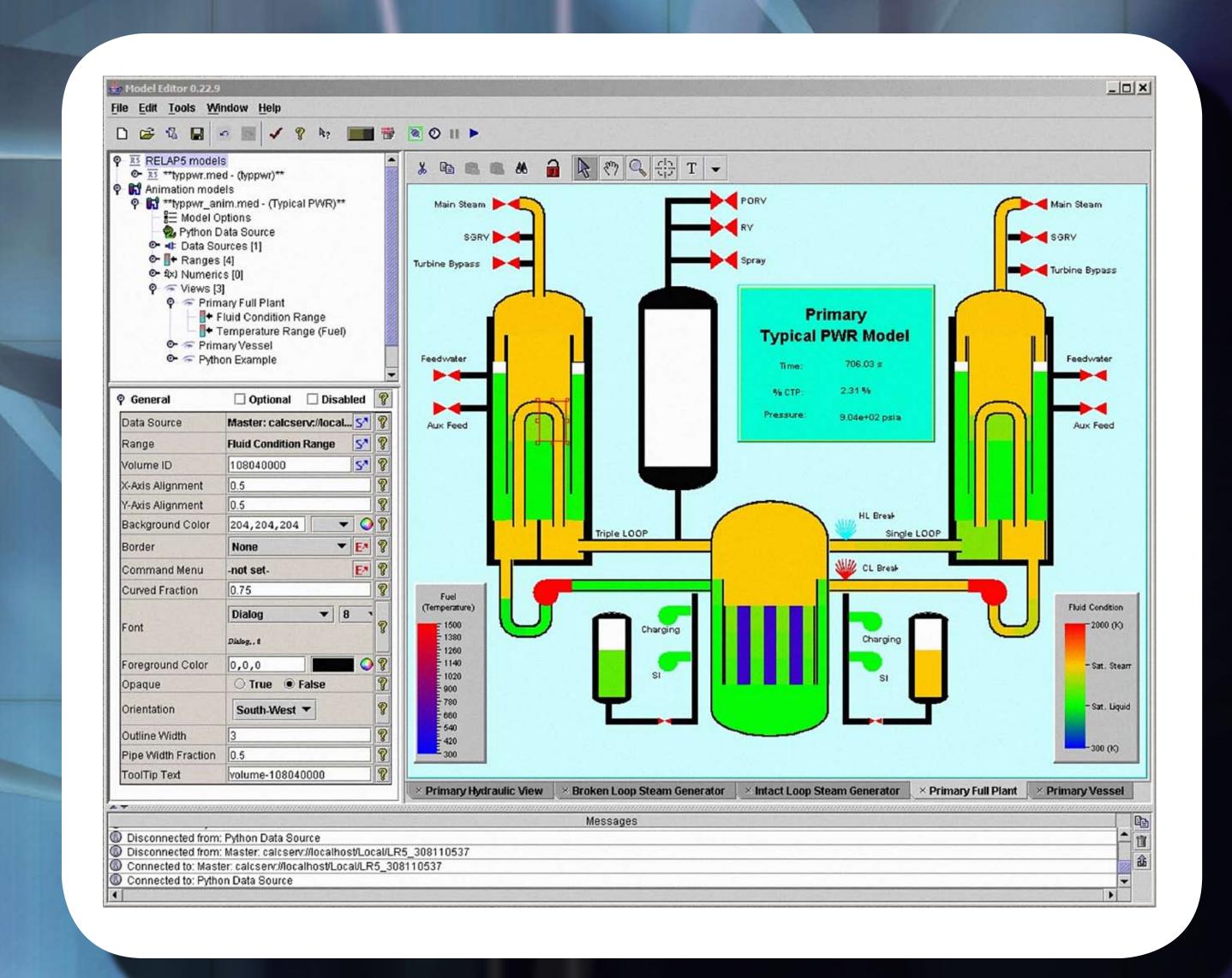


SNAP Execution Monitor

- Most analytical codes need several data files for input and produce many output files for each analysis being done.
- Multiple files can lead to confusion and organization problems for the analyst.
- SNAP Execution Monitor can submit and track jobs so that file clutter is avoided.
- Jobs can even be run and tracked on multiple machines in a networked environment.

SNAP Visualization Tools

- Output from the analytical codes is usually in the form of large ASCII or binary files.
- SNAP provides custom data and model animation features
- SNAP also provides an advanced publication quality plotting tool so that results can be easily used in printed reports.



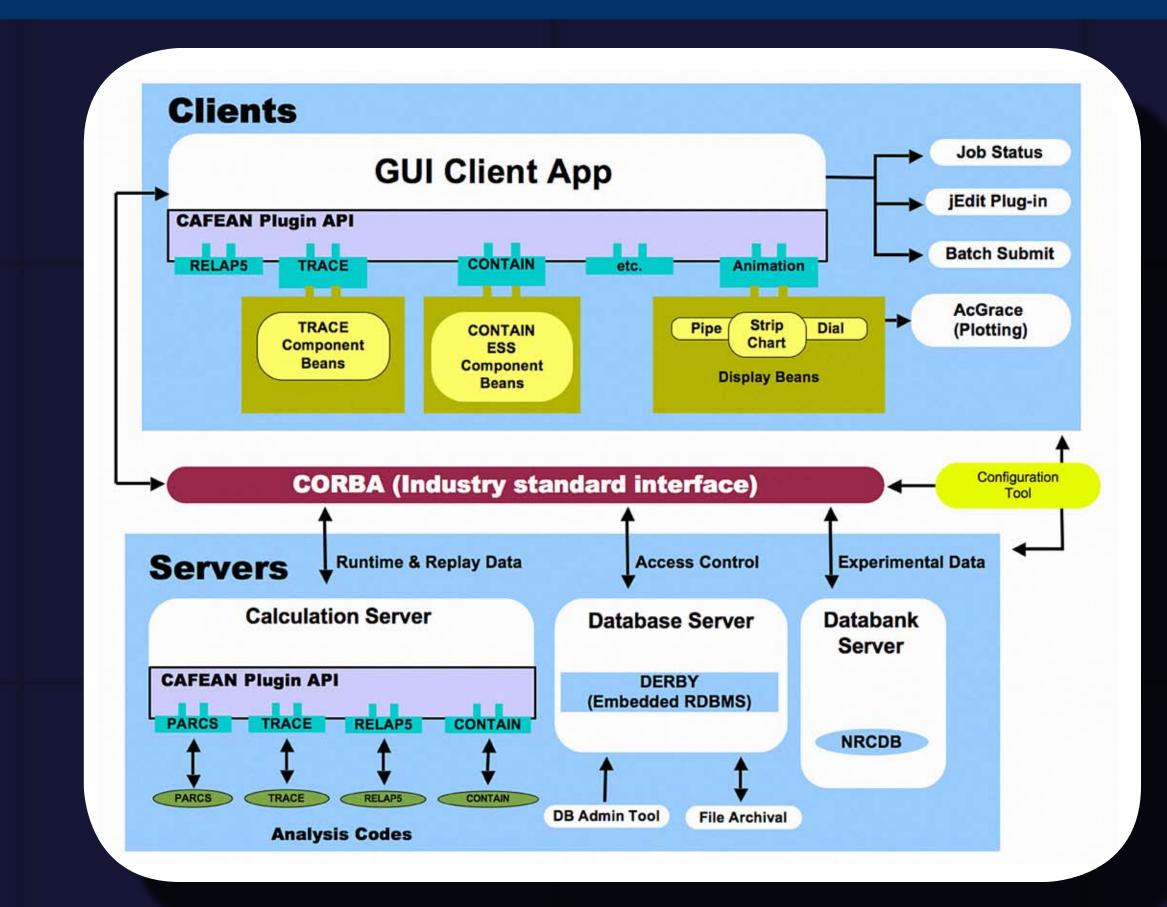
SNAP Design and Programming Features

How it Works

- SNAP is written in Java and makes extensive use of object oriented programming features.
 - Java is supported on most popular computer platforms.
- SNAP uses a "plug-in" design a standard and very popular code design.
 - Permits easy extensibility to other analytical codes.
- SNAP uses industry standard interfaces whenever possible.
 - Reduces maintenance costs.
 - Allows SNAP to make use of third party libraries and features (such as SVG and PDF file format and Derby embedded database).

Extensibility

- CORBA Used for All System Component Public Interfaces
 - Strictly Defined, Well Documented Interfaces
 - **Easy to Implement New Client Apps**
- Plug-in Design Used in Client GUI and Calculation Server
 - Analysis Code Plug-ins
 - Feature Plug-ins
- JavaBean Component Design
 - Shared Repository for Contributed Beans
 - Custom Beans can be independently developed
- Python Scripting
 - User Defined Functions Model Input Calculations
 - Python Data Channels Animations



Plug-ins

- Standard API for Adding New Analysis Codes and New Features
- Plug-in Contains All Analysis Code Specific Classes, I/O, Dialogs, etc...
- No Modification of the Base Code Required
- Current SNAP Code Plug-ins
 - * RELAP5
 - * TRACE
 - CONTAIN
 - ***** FRAPCON3
 - MELCOR
 - PARCS
- **SNAP Feature Plug-ins**
 - "Wizards" can be built in plug-in form, anyone can add a wizard for any purpose without impacting the rest of the SNAP application
- Plug-in API documentation available permitting third party plugin development without impact on other SNAP code users or developers.

JavaBean Plug-ins

- The SNAP plug-ins themselves are built upon the industry standard JavaBean design
- All model components and sub-components are JavaBean compliant objects, complete with BeanInfo classes and property editors.
- Consistent Property Views
- Provides an Intuitive User Interface
- Custom Editing Dialogs
- Multi-selection Editing

